Part 3: Model Performance Report

Data Preparation

The dataset was processed using various preprocessing steps to ensure data quality and optimize it for different models. Duplicate rows were removed to avoid repetition, and missing values in tumor size and inv-nodes were filled with their respective medians to keep important information while reducing the effect of outliers. Categorical intervals within numerical features like age, tumor-size, and inv-nodes were converted to their midpoints to maintain numerical relationships. Other categorical variables such as menopause, breast, and breast-quad were one-hot encoded for better model interpretability. Since K-Nearest Neighbors (KNN) depends on distance calculations, numerical features were scaled using StandardScaler to ensure no single feature has too much influence. This process transforms features to have a mean of zero and a standard deviation of one. This allows the model to treat all numerical variables equally during distance computations.

Data Preparation Insights

The dataset had an uneven distribution of classes, which could make it harder for the model to correctly predict the smaller class and potentially lead to biased predictions. To address this, feature scaling was applied, which significantly improved KNN's performance by ensuring all features contributed equally to distance calculations. It prevents those with larger numerical values from dominating. Filling in missing values allowed to retain valuable data, making the dataset more complete and reliable rather than weakening it by removing entire rows. Additionally, one-hot encoding was used to convert categorical variables into a more machine friendly format, ensuring the model could interpret them correctly without introducing unfair biases. These preprocessing steps helped improve model accuracy and stability, making the predictions more reliable.

Model Training Procedure

Three models were trained to predict recurrence events. First, a basic K-Nearest Neighbors (KNN) model with k=5 was used, benefiting from standardized features to ensure fair distance calculations. To improve its performance, KNN was fine-tuned using Grid Search with 5-fold Cross-Validation, testing different values of k from 1 to 19 (only odd numbers) to find the best one for the dataset. This optimization helped balance bias and variance for better predictions. Lastly, a Support Vector Machine (SVM) with a linear kernel was trained, offering a strong ability to handle complex relationships in the data while maintaining interpretability.

Model Performance Evaluation

The models were tested using accuracy, precision, recall, and F1-score to see how well they predicted recurrence events. The basic KNN model did a decent job but struggled with class imbalance, meaning it had trouble identifying recurrence cases. To improve this, KNN was fine-tuned using Grid Search with Cross-Validation, which slightly boosted accuracy but still had difficulty predicting recurrence cases. The Support Vector Machine (SVM) with a linear kernel performed the best, achieving the highest recall for recurrence cases. This is especially important in medical situations where missing a recurrence could have serious consequences. Since catching all recurrence cases was the main goal, recall was prioritized over overall accuracy. While all models had their pros and cons, SVM provided the best balance between recall and overall performance. The classification reports for each model are shown in the table below:

Model Performance Summary

Model	**Precision (No)**	**Recall (No)**	**F1-score (No)**	**Precision (Yes)**	**Recall (Yes)**	**F1-score (Yes)**	**Accuracy**
KNN (Baseline, k=5)	0.80	0.88	0.84	0.36	0.24	0.29	0.74
KNN (Grid Search CV)	0.79	0.98	0.87	0.50	0.06	0.11	0.78
Logistic Regression	0.81	0.98	0.89	0.75	0.18	0.29	0.81

Model Confidence & Conclusion

The models did a decent job overall due to solid data cleaning, scaling, and cross-validation. But there were still some issues. KNN struggled with class imbalance and did not perform well when dealing with a lot of different features. Even after tuning, it still had trouble correctly predicting recurrence cases. On the other hand, the SVM model performed the best, especially in catching recurrence cases, which is important in a medical setting where missing one could have serious consequences.

Future Improvements

There are a few ways to make these models even better. One big improvement could be using some technique that helps balance out the data so the model gets better at predicting the less common recurrence cases. We could also look for new ways to combine or transform features to help the model find better patterns. Trying out more advanced models could boost accuracy and make predictions more reliable. Finally, tweaking SVM’s settings, like adjusting its regularization, could fine-tune performance even more. Making these changes could help create a stronger model that gives doctors better insights for decision-making.